Supplementary Data – Bioconductor Vignette for DNAshapeR Package

DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding

Tsu-Pei Chiu[1,#], Federico Comoglio[2,#,+], Tianyin Zhou[1,&], Lin Yang[1], Renato Paro[2,3] and Remo Rohs[1,*]

[1]Molecular and Computational Biology Program, Departments of Biological Sciences, Chemistry, Physics, and Computer Science, University of Southern California, Los Angeles, CA 90089, USA
[2]Department of Biosystems Science and Engineering, ETH Zürich, Mattenstrasse 26, 4058 Basel, Switzerland
[3]Faculty of Science, University of Basel, Klingelbergstrasse 50, 4056 Basel, Switzerland
[#]The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors listed in alphabetical order.
[+]Present address: Cambridge Institute for Medical Research, Wellcome Trust/MRC Stem Cell Institute, University of Cambridge, Cambridge CB2 0XY, UK
[&]Present address: Google Inc., 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA
*To whom correspondence should be addressed: rohs@usc.edu

Introduction

DNAshapeR predicts DNA shape features in an ultra-fast, high-throughput manner from genomic sequencing data. The package takes either nucleotide sequence(s) or genomic intervals as input, and generates various graphical representations for further analysis. DNAshapeR further encodes DNA sequence and shape features for statistical learning applications by concatenating feature matrices with user-defined combinations of *k*-mer and DNA shape features that can be readily used as input for machine learning algorithms.

In this vignette, you will learn:

- how to load/install DNAshapeR
- how to predict DNA shape features
- how to visualize DNA shape predictions
- how to encode sequence and shape features, and apply them

Load DNAshapeR

```
library(DNAshapeR)
```

Predict DNA shape features

The core of DNAshapeR, the DNAshape method (Zhou, et al., 2013), uses a sliding pentamer window where structural features unique to each of the 512 distinct pentamers define a vector of minor groove width (MGW), Roll, propeller twist (ProT), and helix twist (HelT) at each nucleotide position. MGW and ProT define base-pair parameters whereas Roll and HelT represent base pair-step parameters. The values for each DNA shape feature as function of its pentamer sequence were derived from all-atom Monte Carlo simulations where DNA structure is sampled in collective and internal degrees of freedom in combination with explicit sodium counter ions (Zhang, et al., 2014). The Monte Carlo simulations were analyzed with a modified Curves approach (Zhou, et al., 2013). Average values of each shape feature for each pentamer were derived from analyzing the ensemble of Monte Carlo predictions for 2,121 DNA fragments of 12–27 base pairs in length. DNAshapeR predicts the four DNA shape features MGW, HelT, ProT, and Roll, which were observed in various cocrystal structures as playing an important role in specific protein-DNA binding.

DNAshapeR can predict DNA shape features from custom FASTA files or directly from genomic coordinates in the form of a GRanges object within Bioconductor (see https://bioconductor.org/packages/release/bioc/html/GenomicRanges.html for more information).

From FASTA file

To predict DNA shape features from a FASTA file

```
library(DNAshapeR)

fn <- system.file("extdata", "CGRsample.fa", package = "DNAshapeR")

pred <- getShape(fn)

## Reading the input sequence......

## Reading the input sequence......

## Reading the input sequence......

## Reading the input sequence......

## Parsing files......

## Record length: 2000

## Record length: 1999

## Record length: 2000

## Record length: 1999

## Done
```

From genomic intervals (e.g., TFs binding sites, CpG islands, replication origins)

To predict DNA shape from genomic intervals stored as GRanges object, a reference genome is required. Several reference genomes are available within BioConductor as BSgenome objects (see http://bioconductor.org/packages/release/bioc/html/BSgenome.html for more information). For example, the sacCer3 release of the *S.Cerevisiae* genome can be retrieved by

```
# Install Bioconductor packages

source("http://bioconductor.org/biocLite.R")

biocLite("BSgenome.Scerevisiae.UCSC.sacCer3")

library(BSgenome.Scerevisiae.UCSC.sacCer3)
```

Given a reference genome, the **getFasta** function first extracts the DNA sequences based on the provided genomic coordinates, and then performs shape predictions within a user-defined window (of size equal to width, 100 bp in the example below) computed from the center of each genomic interval:

```
# Create a query GRanges object

gr <- GRanges(seqnames = c("chrI"),

              strand = c("+", "-", "+"),

              ranges = IRanges(start = c(100, 200, 300), width = 100))

getFasta(gr, Scerevisiae, width = 100, filename = "tmp.fa")

fn <- "tmp.fa"

pred <- getShape(fn)
```

From public domain projects

The genomic intervals can also be obtained from public domain projects, including ENCODE, NCBI, Ensembl, etc. The AnnotationHub package
(see http://bioconductor.org/packages/release/bioc/html/AnnotationHub.html for more information) provides an interface to retrieve genomic intervals from these multiple online project resources.

```
# Install Bioconductor packages

library(BSgenome.Hsapiens.UCSC.hg19)

library(AnnotationHub)
```

The genomic intervals of interest can be selected progressively through the functions of **sebset** and **query** with keywords, and can be subjected as an input of GRanges object to **getFasta** function.

```
ah <- AnnotationHub()

ah <- subset(ah, species=="Homo sapiens")

ah <- query(ah, c("H3K4me3", "Gm12878", "Roadmap"))

getFasta(ah[[1]], Hsapiens, width = 150, filename = "tmp.fa")

fn <- "tmp.fa"

pred <- getShape(fn)
```

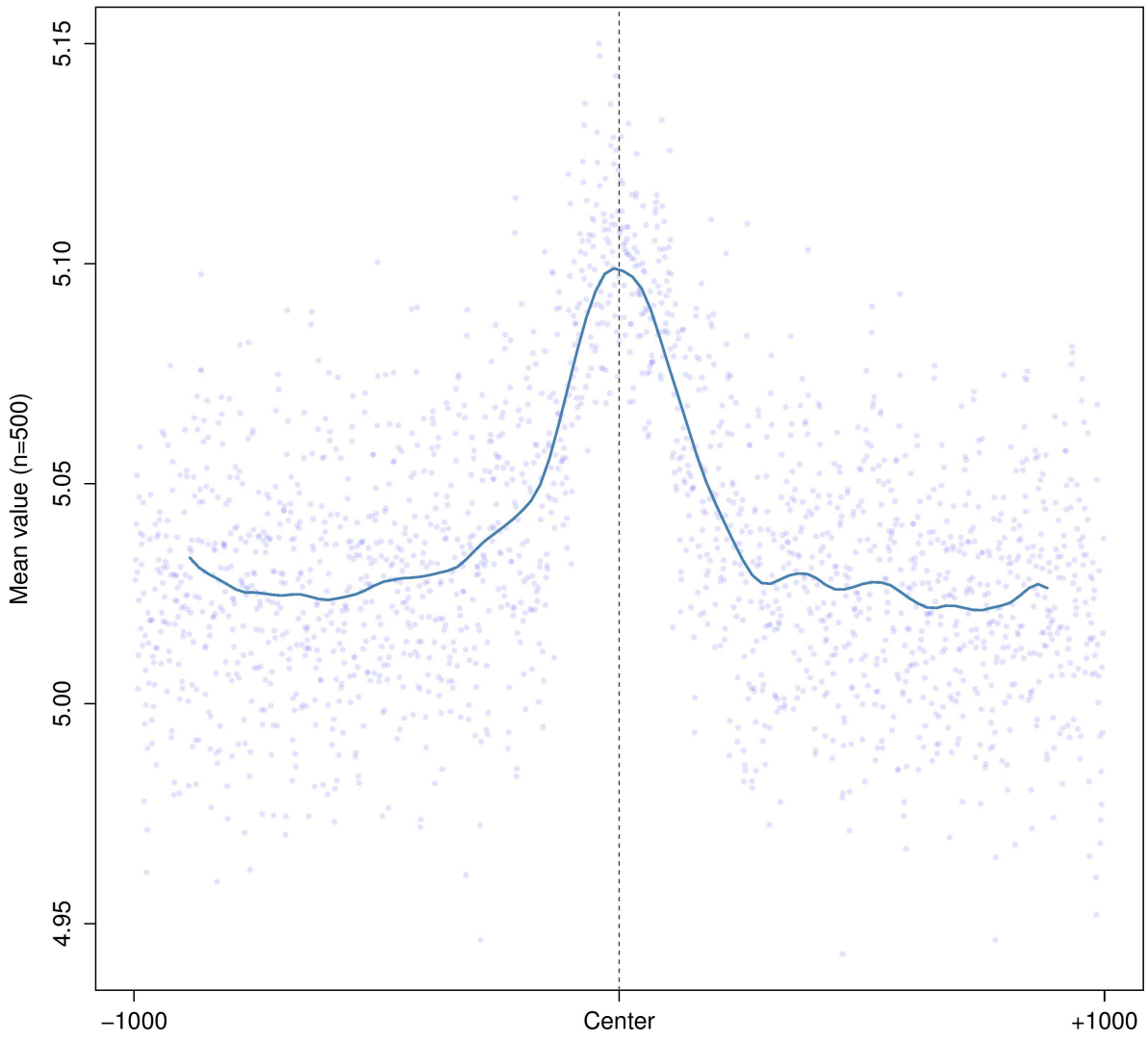
Visualize DNA shape prediction

DNAshapeR can be used to generate various graphical representations for further analyses. The prediction result can be visualized in the form of scatter plots (as introduced in Comoglio, et al., 2015), heat maps (as introduced in Yang, et al., 2014), or genome browser tracks (as introduced in Chiu, et al., 2015).

Ensemble representation: metashape plot

The prediction result can be visualized in the metaprofiles of DNA shape features.

```
plotShape(pred$MGW)
```

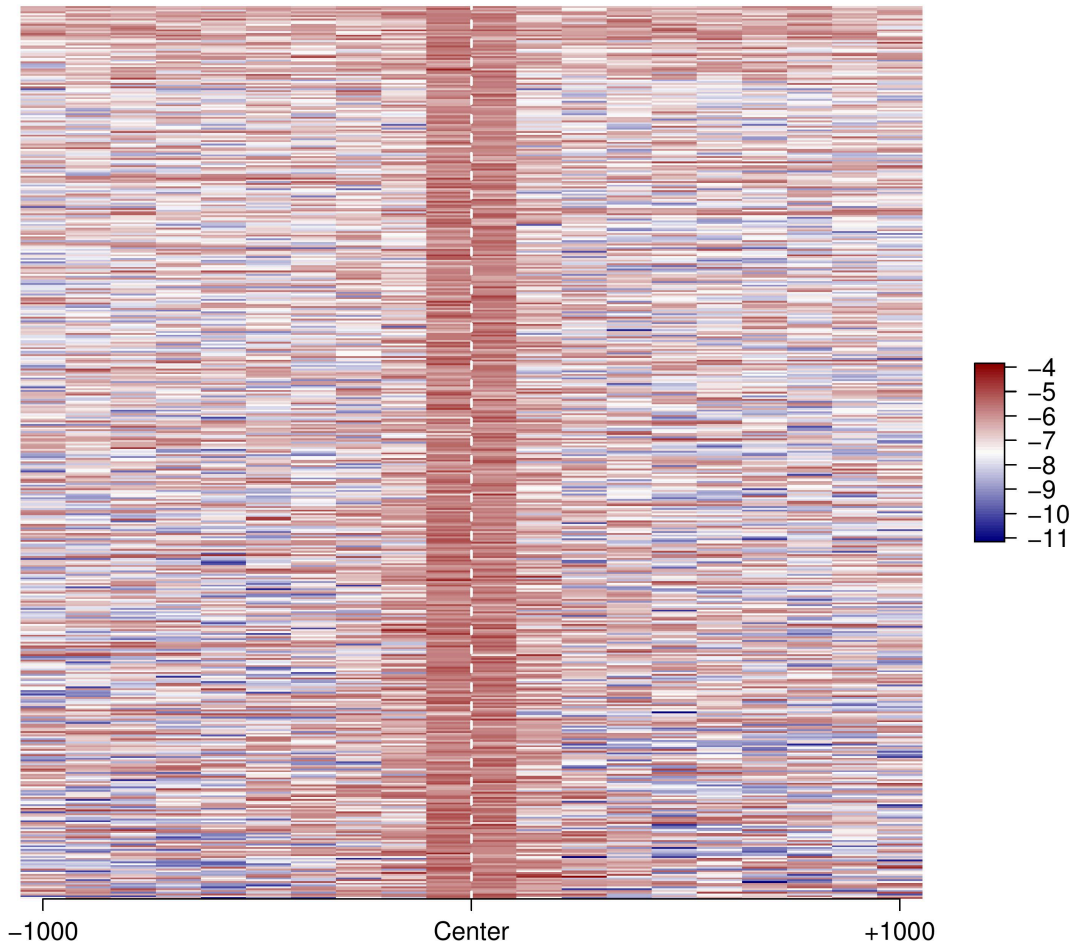



```
#plotShape(pred$ProT)

#plotShape(pred$Roll)

#plotShape(pred$HelT)
```

Ensemble representation: heat map

The prediction result can be visualized in the heat map of DNA shape features.

```
library(fields)

heatShape(pred$ProT, 20)
```



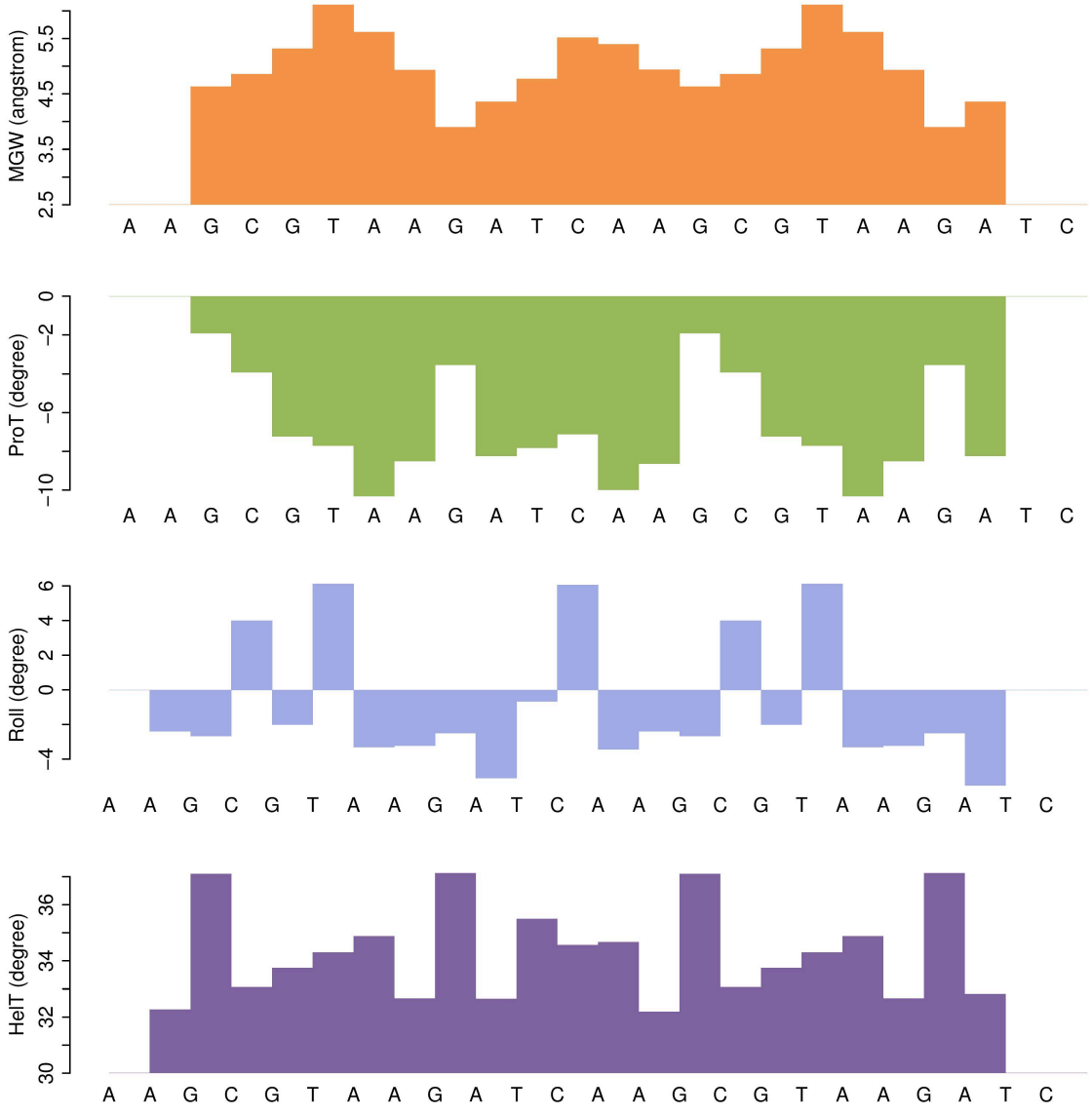

```
#heatShape(pred$MGW, 20)

#heatShape(pred$Roll[1:500, 1:1980], 20)

#heatShape(pred$HelT[1:500, 1:1980], 20)
```

Individual representation: genome browser-like tracks

The prediction result can be visualized in the form of genome browser tracks.

*Note that the input data should only contain one sequence.

```
fn2 <- system.file("extdata", "SingleSeqsample.fa", package = "DNAshapeR")

pred2 <- getShape(fn2)

trackShape(fn2, pred2) # Only for single sequence files
```



Encode sequence and shape features

DNAshapeR can be used to generate feature vectors for a user-defined model. These models can consist of either sequence features (1-mer, 2-mer, 3-mer), shape features (MGW, Roll, ProT, HelT), or any combination of those two. For 1-mer features, sequence is encoded in form of four binary numbers (i.e., in terms of 1-mers, 0001 for adenine, 0010 for cytosine, 0100 for guanine, and 1000 for thymine) at each nucleotide position (Zhou, et al., 2015). The feature encoding function of the DNAshapeR package enables the determination of higher order sequence features, for example, 2-mers and 3-mers (16 and 64 binary features at each position, respectively).

The user can also choose to include second order shape features in the generated feature vector. The second order shape features are product terms of values for the same category of shape features (MGW, Roll, ProT or HelT) at adjacent positions. They were introduced to encode the tendency of, for instance, a narrow minor groove region exhibiting an enhanced narrowing if adjacent positions are also characterized by a narrow groove (Zhou, et al., 2015). The feature encoding function of DNAshapeR enables the generation of any subset of these features, either only a selected shape category or first order shape features, and any combination with shape or sequence features. The result of feature encoding for each sequence is a chimera feature vector.

Encoding process

A feature type vector should be defined before encoding. The vector can be any combination of characters of “k-mer”, “n-shape”, “n-MGW”, “n-ProT”, “n-Roll”, “n-HelT” (k, n are integers) where “1-shape” refers to first order and “2-shape” to second order shape features.

The features of each shape category were normalized to values between 0 and 1 by Min-Max Normalization method. The minimum and maximum values used in the normalization of first order shape features were derived from the pentamer query table (for example, minimum and maximum values of MGW are 2.85 and 6.2 Å, respectively). The second order shape features are the products of normalized first order features. The minimum and maximum values of these products were chosen for the higher order normalization.

```
library(Biostrings)

featureType <- c("1-mer", "1-shape")

featureVector <- encodeSeqShape(fn, pred, featureType)

featureVector
```

The normalization function can be turned off by setting the argument “normalize”.

```
featureVector <- encodeSeqShape(fn, pred, featureType, normalize = FALSE)

featureVector
```

Showcase of statistical machine learning application

Feature encoding of multiple sequences thus results in a feature matrix, which can be used as input for variety of statistical machine learning methods. For example, an application is the quantitative modeling of SELEX-seq derived protein-DNA binding by linear regression as demonstrated below.

First, the experimental binding affinity values are combined with the feature matrix in a data frame structure.

```
filename3 <- system.file("extdata", "SELEXsample.s", package = "DNAshapeR")
```

```
experimentalData <- read.table(filename3)

df <- data.frame(affinity=experimentalData$V1, featureVector)
```

Then, a machine learning package (which can be any learning tools) is used to train a multiple linear regression (MLR) model based on 10-fold cross-validation. In this example, we used the caret package (see http://caret.r-forge.r-project.org/ for more information).

```
library(caret)



trainControl <- trainControl(method = "cv", number = 10, savePredictions = TRUE)

model <- train (affinity~ ., data = df, trControl=trainControl, method="lm", preProcess=NULL)

summary(model)
```

Session Info

```
sessionInfo()

## R version 3.2.2 (2015-08-14)

## Platform: x86_64-w64-mingw32/x64 (64-bit)

## Running under: Windows 7 x64 (build 7601) Service Pack 1

##

## locale:

## [1] LC_COLLATE=English_United States.1252

## [2] LC_CTYPE=English_United States.1252

## [3] LC_MONETARY=English_United States.1252

## [4] LC_NUMERIC=C

## [5] LC_TIME=English_United States.1252

##

## attached base packages:

## [1] grid      stats     graphics  grDevices utils     datasets  methods

## [8] base

##
```

```
## other attached packages:

## [1] fields_8.3-5     maps_3.0.0-2     spam_1.2-1       DNAshapeR_0.99.1

##

## loaded via a namespace (and not attached):

##  [1] magrittr_1.5    formatR_1.2.1   tools_3.2.2     htmltools_0.2.6

##  [5] yaml_2.1.13     Rcpp_0.12.1     stringi_1.0-1   rmarkdown_0.8.1

##  [9] knitr_1.11      stringr_1.0.0   digest_0.6.8    evaluate_0.8
```

Author contributions

T.P.C., F.C., and R.R. designed and executed this project with the help of T.Z., L.Y., and R.P. based on methods developed by T.P.C., T.Z., and L.Y. The project was conceived by F.C. and directed by R.R.

Acknowledgements

The authors acknowledge comments and suggestions by members of the Rohs lab. This work was supported by the NIH (R01GM106056, R01HG003008 in part, and U01GM103804 to R.R.). Release of open-source software and open-access publication were supported by the NSF (MCB-1413539 to R.R.). R.R. is an Alfred P. Sloan Research Fellow.

References

Chiu, T.P., et al. GBshape: a genome browser database for DNA shape annotations. Nucleic Acids Res. 2015;43(Database issue):D103-D109.

Comoglio, F., et al. High-resolution profiling of Drosophila replication start sites reveals a DNA shape and chromatin signature of metazoan origins. Cell Rep. 2015;11(5):821-834.

Yang, L., et al. TFBSshape: a motif database for DNA shape features of transcription factor binding sites. Nucleic Acids Res. 2014;42(Database issue):D148-155.

Zhou, T., et al. Quantitative modeling of transcription factor binding specificities using DNA shape. Proc. Natl. Acad. Sci. U S A 2015;112(15):4654-4659.

Zhou, T., et al. DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale. Nucleic Acids Res. 2013;41(Web Server issue):W56-62.

Zhang, X., et al. Conformations of p53 response elements in solution deduced using site-directed spin labeling and Monte Carlo sampling. Nucleic Acids Res. 2014;42(4):2789-2797.